

关注微信公众号: PMvideo

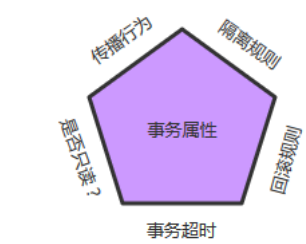
【Spring学习33】 Spring事务(3): 事务属性之隔离级别(isolation level)

作者: soonfly (/authorarticle.html?author=soonfly) 2017-04-20 ☆ 收录到我的专题 (/select.html?articleId=401733)

标签 事务 (<http://www.bijishequ.com/info/search.html?searchText=事务>) 读取 (<http://www.bijishequ.com/info/search.html?searchText=读取>) 隔离 (<http://www.bijishequ.com/info/search.html?searchText=隔离>) 级别 (<http://www.bijishequ.com/info/search.html?searchText=级别>) 数据 (<http://www.bijishequ.com/info/search.html?searchText=数据>)

事务属性

上文的例子中,在声明事务时,用到了 `@Transactional(readonly = false, propagation = Propagation.REQUIRED)`。中间的参数 `readonly`, `propagation` 我们称为事务属性。它就是对事务的基本配置。事务属性有五个方面: 传播行为, 隔离级别, 事务超时时间, 回滚规则, 是否只读。



由属性接口 `TransactionDefinition` 可以看到,可返回四个基本事务属性:

```
1 public interface TransactionDefinition {
2     int getPropagationBehavior(); // 传播行为。
3     int getIsolationLevel(); // 隔离级别。事务管理器根据它来控制另外一个事务可以看到本事务内的哪些数据。
4     int getTimeout(); // 事务必须在多少秒内完成。
5     boolean isReadOnly(); // 事务是否只读。事务管理器能够根据这个返回值进行优化, 确保事务是只读的
6 }
```

这篇先谈事务的隔离级别 (isolation level)

数据库是要被广大客户所共享访问的,那么在数据库操作过程中很可能出现以下几种不确定情况。

更新丢失(Lost update)
两个事务都同时更新一行数据,一个事务对数据的更新把另一个事务对数据的更新覆盖了。比如CMS系统中,两个同时打开一篇文章进行修改,一个人先保存,另一个人后保存,后保存的就覆盖了先保存的那个人的内容,这就造成更新丢失。
这是因为系统没有执行任何的锁操作,因此并发事务并没有被隔离开来。在并发事务处理带来的问题中,“更新丢失”通常应该是完全避免的。但防止更新丢失,并不能单靠数据库事务控制器来解决,需要应用程序对要更新的数据加必要的锁来解决,因此,防止更新丢失应该是应用的责任。

脏读 (Dirty reads)
一个事务读取到了另一个事务未提交的数据操作结果。这是相当危险的,因为很可能所有的操作都被回滚。

不可重复读 (Non-repeatable Reads)
一个事务对同一行数据重复读取两次,但是却得到了不同的结果。比如事务T1读取某一数据后,事务T2对其做了修改,当事务T1再次读该数据时得到与前一次不同的值。又叫虚读。

幻读 (Phantom Reads)
事务在操作过程中进行两次查询,第二次查询的结果包含了第一次查询中未出现的数据或者缺少了第一次查询中出现的数据 (这里并不要求两次查询的SQL语句相同)。这是因为在两次查询过程中有另外一个事务插入数据造成的。

不可重复读的重点是修改某个记录字段,幻读的重点在于新增或者删除记录。
对于前者,只需要锁住满足条件的记录。对于后者,要锁住满足条件及其相近的记录。

“脏读”、“不可重复读”和“幻读”,其实都是数据库读一致性问题,必须由数据库提供一定的事务隔离机制来解决。

事务隔离级别

为了避免上面出现的几种情况，在标准SQL规范中，定义了4个事务隔离级别，由低到高依次为Read uncommitted、Read committed、Repeatable read、Serializable，这四个级别可以逐个解决脏读、不可重复读、幻读这几类问题。

	读数据一致性	脏读	不可重复读	幻读
未提交读 Read uncommitted	最低级别，只能保证不读取物理上损坏的数据	√	√	√
已提交读 Read committed	语句级	×	√	√
可重复读 Repeatable read	事务级	×	×	√
可序列化 Serializable	最高级别，事务级	×	×	×

未提交读取（Read Uncommitted）
Spring标识：ISOLATION_READ_UNCOMMITTED。允许脏读取，但不允许更新丢失。如果一个事务已经开始写数据，则另外一个事务则不允许同时进行写操作，但允许其他事务读此行数据。该隔离级别可以通过“排他写锁”实现。

已提交读取（Read Committed）
Spring标识：ISOLATION_READ_COMMITTED。允许不可重复读取，但不允许脏读取。这可以通过“瞬间共享读锁”和“排他写锁”实现。读取数据的事务允许其他事务继续访问该行数据，但是未提交的写事务将会禁止其他事务访问该行。

可重复读取（Repeatable Read）
Spring标识：ISOLATION_REPEATABLE_READ。禁止不可重复读取和脏读取，但是有时可能出现幻读数据。这可以通过“共享读锁”和“排他写锁”实现。读取数据的事务将会禁止写事务（但允许读事务），写事务则禁止任何其他事务。

序列化（Serializable）
Spring标识：ISOLATION_SERIALIZABLE。提供严格的事务隔离。它要求事务序列化执行，事务只能一个接着一个地执行，不能并发执行。仅仅通过“行级锁”是无法实现事务序列化的，必须通过其他机制保证新插入的数据不会被刚执行查询操作的事务访问到。

隔离级别越高，越能保证数据的完整性和一致性，但是对并发性能的影响也越大。对于多数应用程序，可以优先考虑把数据库系统的隔离级别设为Read Committed。它能够避免脏读取，而且具有较好的并发性能。尽管它会导致不可重复读、幻读和第二类丢失更新这些并发问题，在可能出现这类问题的个别场合，可以由应用程序采用悲观锁或乐观锁来控制。

Spring中同时提供一个标识：ISOLATION_DEFAULT。表示使用后端数据库默认的隔离级别。大多数数据库默认的事务隔离级别是Read committed，比如Sql Server , Oracle。MySQL的默认隔离级别是Repeatable read。



登录

/login.htm

?redirectUrl=%2Fdetail%2F401733)

后发表评论

0条评论或问题

社区邀请

笔记社区是一个面向中高端IT开发者、程序员的知识共享社区，通过网络抓取与文章分类总结，由专家为用户提供高质量的专题文章系列。

 邀请您成为社区专家 >> (<http://www.bijishequ.com/creat.html>)

原文链接：<http://blog.csdn.net/soonfly/article/details/70305585>

声明：所有文章资源均从网络抓取，如果侵犯到您的著作权，请联系删除文章。联系方式请关注微信公众号PMvideo【锤子视频-程序员喜欢的短视频】，或者加笔记社区开发者交流群 628286713。

4 签到

今日签到1人

January

相关标签

- Struts (/tag/list?tagId=1440)
- loc (/tag/list?tagId=1441)
- Spring (/tag/list?tagId=1438)
- SpringMVC (/tag/list?tagId=1560)
- 分布式事务 (/tag/list?tagId=1285)
- Group Commit (/tag/list?tagId=1286)
- 成组提交 (/tag/list?tagId=1283)
- undo (/tag/list?tagId=1282)
- 隔离级别 (/tag/list?tagId=1284)
- BufferedInputStream (/tag/list?tagId=72)
- Level (/tag/list?tagId=441)
- GridData (/tag/list?tagId=442)

- Target (/tag/list?tagId=75)
- TextArea (/tag/list?tagId=443)
- Scheme (/tag/list?tagId=440)
- NoSuchAlgorithmException (/tag/list?tagId=73)
- ClassPathXmlApplicationContext (/tag/list?tagId=439)
- Executors (/tag/list?tagId=76)
- 快照读 (/tag/list?tagId=1290)
- 幻读 (/tag/list?tagId=1293)
- 脏读 (/tag/list?tagId=1291)
- 重复读 (/tag/list?tagId=1292)
- 行锁 (/tag/list?tagId=1294)
- 存储引擎 (/tag/list?tagId=1745)
- 数据库 (/tag/list?tagId=1743)
- 表格系统 (/tag/list?tagId=1744)
- OceanBase (/tag/list?tagId=1741)
- NoSql (/tag/list?tagId=1742)
- 锁定读 (/tag/list?tagId=1289)
- Mysql事务 (/tag/list?tagId=1279)
- Mysql索引 (/tag/list?tagId=1270)
- Mysql锁 (/tag/list?tagId=1287)
- Mysql (/tag/list?tagId=1746)
- J2EE (/tag/list?tagId=2397)
- 编程语言 (/tag/list?tagId=2761)
- java (/tag/list?tagId=2396)
- Mysql备份与恢复 (/tag/list?tagId=1297)
- 云存储 (/tag/list?tagId=1730)
- 数据存储 (/tag/list?tagId=1729)

相关文章

- Spring源码分析之AOP解析 (/detail/577731?p=)
- Spring事务管理入门与进阶 (/detail/577593?p=)
- 通过注解实现Spring 声明式事务管理 (/detail/577865?p=)
- spring boot 集成shiro详细教程 (/detail/578006?p=)
- SpringIOC源码阅读—BeanDefinitionParserDelegate (/detail/577695?p=)
- SpringIOC源码阅读—解析xml创建BeanDefinition (/detail/577690?p=)
- Spring BeanPostProcessor接口使用 (/detail/577929?p=)
- Spring Aware相关接口的使用 (/detail/577831?p=)
- SpringIOC源码阅读—BeanDefinitionDocumentReader (/detail/577693?p=)
- 【编码日常】如何修改动态代理的私有变量 (/detail/577591?p=)

相关专题



SpringBoot学习笔记

Spring Boot是由Pivotal团队提供的全新框架，其设计目的是用来简化新Spring应用的初始搭建以及开发过程。

共 0 篇文章

(/subject/219)



springboot

文件上传在各种网站平台上应用都非常广泛，这篇文章将讲述在Springboot中是如何完成文件上传的，Springboot是打包运行的，上传后的文件又该何去何从？

共 4 篇文章

(/subject/408)



Springboot项目笔记

用spring boot 已经写过一个项目了，第二个项目在搭建框架的时候还是优点懵逼。到处查资料，尤其是数据链接那块，所以成现在有时间还是记录下。

共 3 篇文章

(/subject/261)



Spring boot

Spring Boot开发：从0到1

共 1 篇文章

(/subject/453)



spring框架

设计模式有助于遵循良好的编程实践。作为最流行的Web框架之一的Spring框架也使用其中的一些。

共 3 篇文章

(/subject/277)



spring cloud

Spring Cloud集成相关优质项目推荐 这些项目是Spring Cloud官方项目或是对Spring Cloud进行了有益的补充以及基于Spring Cloud最佳实践。

共 0 篇文章

(/subject/287)



Spring核心技术

Spring框架对于IoC规则的实现。IoC也同DI(依赖注入)。而对象是通过构造函数，工厂方法，或者一些Set方法来定义对象之间的依赖的。

共 15 篇文章

(/subject/331)




springboot框架搭建学习与实例

这是我学习SpringBoot的入门系列哦，欢迎阅读。SpringBoot的推出不是用来替代Spirng的，而是精简Spring的配置的。也就是优化Spirng的。

共 11 篇文章

(/subject/11)



springBoot RabbitMQ

全面介绍springBoot RabbitMQ

共 0 篇文章

(/subject/336)



Springboot详细入门教程

Spring Boot是由Pivotal团队提供的全新框架，其设计目的是用来简化新Spring应用的初始搭建以及开发过程。

共 14 篇文章

(/subject/179)

精选专题	
<div><div>spring系列 (/info/spring-series.html)</div><div>1893篇</div></div>	<div><div>ORM (/info/orm.html)</div><div>1622篇</div></div>
<div><div>缓存 (/info/cache.html)</div><div>2346篇</div></div>	<div><div>消息中间件 (/info/mq.html)</div><div>534篇</div></div>
<div><div>分布式服务 (/info/ds.html)</div><div>1792篇</div></div>	<div><div>Nio框架 (/info/nio.html)</div><div>691篇</div></div>
<div><div>并发编程 (/info/concurrent.html)</div><div>2110篇</div></div>	<div><div>网络基础 (/info/network.html)</div><div>18925篇</div></div>
<div><div>搜索引擎 (/info/searchEngine.html)</div><div>962篇</div></div>	<div><div>设计模式 (/info/design-mode.html)</div><div>2941篇</div></div>
<div><div>其他arch (/info/other-arch.html)</div><div>2837篇</div></div>	<div><div>计算机基础 (/info/computer.html)</div><div>2885篇</div></div>
<div><div>操作系统 (/info/os.html)</div><div>11.2篇</div></div>	<div><div>开发工具 (/info/developer.html)</div><div>3575篇</div></div>